RECOMMENDATION ITU-R BT.1676

Methodological framework for specifying accuracy and cross-calibration of video quality metrics

(Question ITU-R 44/6)

(2004)

The ITU Radiocommunication Assembly,

considering

a) that digital TV and HDTV utilizing bit-rate reduction technologies such as MPEG-2, DV and others have achieved widespread use;

b) that the Radiocommunication Sector is responsible for setting the overall quality performance of broadcasting chains;

c) that impairments to television pictures can be shown to correlate with measurable features of the signals;

d) that overall picture quality is related to the combination of all impairments;

e) that in the case of digital TV it is necessary, in particular, to assess the performance of bit-rate reduction methods both in terms of subjective and objective parameters;

f) that for television systems a number of objective picture quality parameters as well as associated performance measurement and monitoring methods have been developed for the studio environment and in broadcasting;

g) that full reference objective picture quality measurement methods are useful in evaluating studio and broadcasting systems;

h) that data sets of test materials, subjective scores and objective values are used in the validation testing of objective picture quality measurement methods;

j) that there are a number of proposed full reference video quality metrics (VQMs) that can be used to provide objective picture quality ratings;

k) that there are a number of well-known statistical evaluation methods documented in the literature that can be used to validate and compare VQMs based on data sets of test materials, subjective scores and objective values;

l) that when one or more VQMs are accepted as normative in ITU Recommendations there will still be a need to estimate the mathematical accuracy (resolving power) of the VQM being used;

m) that cross-calibration of full reference objective picture quality measurement methods based on available data sets is important for international exchange of measurement and monitoring results,

recommends

1 that the calculations specified in Annex 1 be used to estimate the accuracy and cross-calibration of objective picture quality measurements utilizing the full reference method;

2 that the calculations specified in Annex 1 may be used as one of several methods to determine the accuracy in evaluation and validation of various objective picture quality measurements utilizing the full reference method.

Annex 1

Method for specifying accuracy and cross-calibration of VQMs

1 Scope

VQMs are intended to provide calculated values that are strongly correlated with viewer subjective assessments. This Recommendation provides:

– Methods for curve fitting VQM objective values to subjective data in order to facilitate the accuracy calculation and to produce a normalized objective value scale that can be used for cross correlation between different VQMs.

– An algorithm (based on statistical analysis relative to subjective data) to quantify the accuracy of a given VQM.

– A simplified root mean square error calculation to quantify the accuracy of a VQM when the subjective data has roughly equal variance across the VQM scale.

– A method to plot classification errors to determine the relative frequencies of “false tie”, “false differentiation”, “false ranking”, and “correct decision” for a given VQM.

The methods specified in this Recommendation are based on objective and subjective evaluation of component video such as defined by Recommendation ITU-R BT.601 using methods such as described in Recommendation ITU-R BT.500 – Methodology for the subjective assessment of the quality of television pictures. A data set for a VQM will consist of objective values and mean subjective scores for a variety of motion video sources (SRCs) processed by a variety of hypothetical reference circuits (HRCs). An example of such a data set is given in ITU-T Document COM 9-80-E, Final report from the video quality experts group on the validation of objective models of video quality assessment.

The methods specified in this Recommendation are directly applicable to a defined data set as described above. For measurements not specifically part of the data set the methods specified in this Recommendation provide a reasonable estimate of accuracy and cross-calibration for applications that can be considered to be similar to and within the scope of the defined data set.

The methods specified in this Recommendation are appropriate for use in combination with other statistical calculations in order to evaluate the usefulness of a VQM. Informative information regarding the use of the methods is presented in Appendix 1. A complete verification process by suitable independent laboratories is required for a VQM to be considered for inclusion as a normative part of an ITU-R Recommendation.

2 Accuracy of a VQM

In order to use an objective VQM, one must know whether the score difference between two processed videos is statistically significant. Hence, a quantification is needed of the accuracy (or resolving power) of the VQM. To visualize this resolving power, it helps to begin with a scatter plot in which the abscissa of each point is a VQM score from a particular video SRC and distortion HRC, and the ordinate is a subjective score from a particular viewing of the SRC/HRC. Each SRC/HRC combination (associated with a particular VQM score) contains a distribution of mean subjective scores, S, based on a number of viewers, which represents (approximately) the relative probabilities of S for the particular SRC/HRC combination. The resolving power of a VQM can be defined as the difference between two VQM values for which the corresponding subjective-score distributions have means that are statistically different from each other (typically at the 0.95% significance level).

Given this qualitative picture, two metrics for resolving power will be described in this section, each one useful in a different context. The metrics are described in § 2.3 and 2.4. Also, in § 2.5, a method is described for evaluating the frequencies of different kinds of errors made by the VQM. As an example of implementation of all the methods, a computer source code in MATLAB (The Mathworks, Inc., Natick, MA) is provided in Appendix 2.

2.1 Nomenclature and coordinate scales

Let each SRC/HRC combination in a data set be called a "situation", and let N be the number of situations in this data set. A subjective score for situation i and viewer l will be denoted as S_{il}, and an objective score for situation i will be denoted as O_i. Averaging over a variable such as viewer will be denoted with a dot in that variable location. For instance, the mean opinion score of a situation will be denoted as $S_{i\bullet}$. The subjective-score statistics from each pair (i, j) of these situations are to be assessed for significance of VQM difference, and then used to arrive at a resolving power for the VQM difference, as a function of the VQM value.

Prior to any statistical analysis, the original subjective mean opinion scores $S_{i\bullet}$ are linearly transformed to the interval [0, 1], defined as the Common Scale, where 0 represents no impairment and 1 represents most impairment. If best represents the no-impairment value of the original subjective score and worst represents the maximum impairment of the original subjective scale, then the scaled scores $\hat{S}_{i\bullet}$ are given by:

$$\hat{S}_{i\bullet} = \frac{S_{i\bullet} - best}{word - best}$$

Next, the VQM scores are transformed to this common scale as a by-product of the process of fitting the VQM scores to the subjective data, which will be discussed in the following section.

2.2 Fitting VQM values to subjective data

Fitting removes systematic differences between the VQM and the subjective data (e.g. d.c. shift) that do not provide any useful quality discrimination information. In addition, fitting all VQMs to one common scale will provide a method for cross-calibration of those VQMs.

The simplest method of data fitting is linear correlation and regression. For subjective video quality scores, this may not be the best method. Experience with other video quality data sets indicates chronically poor fits of VQM to subjective scores at the extremes of the ranges. This problem can be ameliorated by allowing the fitting algorithm to use non-linear, but still monotonic (order-preserving), methods. If a good non-linear model is used, the objective-to-subjective errors will be smaller and have a central tendency closer to zero.

Non-linear methods can be constrained to effectively transform the VQM scale to the [0, 1] common scale. Besides improving the fit of data with a VQM, a fitting curve also offers an additional advantage over the straight-line fit implied by the Native Scale (i.e. the original scale of the VQM): the distribution of objective-to-subjective errors around the fitted model curve is less dependent on the VQM score. Of course, the non-linear transformation may not remove all the score dependency of objective-to-subjective errors. To capture the residual dependence, it would ideally have been useful to record objective-to-subjective error as a function of VQM value. However, typical data sets are too small to divide among VQM bins in a statistically robust way. Therefore, as will be clear in § 2.3, a sort of average measure over the VQM range is computed.

Figure 1 shows the improved fit of model to data incurred by transforming the objective scores using a fitting function. It can be seen that, besides improving the fit of data with VQM, the curve also offers an additional advantage over the straight-line fit implied by the native scale: the distribution of model-to-data errors around the fitted model curve is less dependent on the VQM score.

We denote the original (native scale) objective scores O_i, and the common scale objective scores as $\hat{O}_i$. A fitting function F (depending on some fitting parameters), connects the two. The function used to fit the objective VQM data (O_i) to the scaled subjective data ($\hat{S}_{i\bullet}$) must have the following three attributes:

– a specified domain of validity, which should include the range of VQM data for all the situations used to define the accuracy metric;

– a specified range of validity, defined as the range of common scale scores (a sub-range of [0, 1]) to which the function maps;

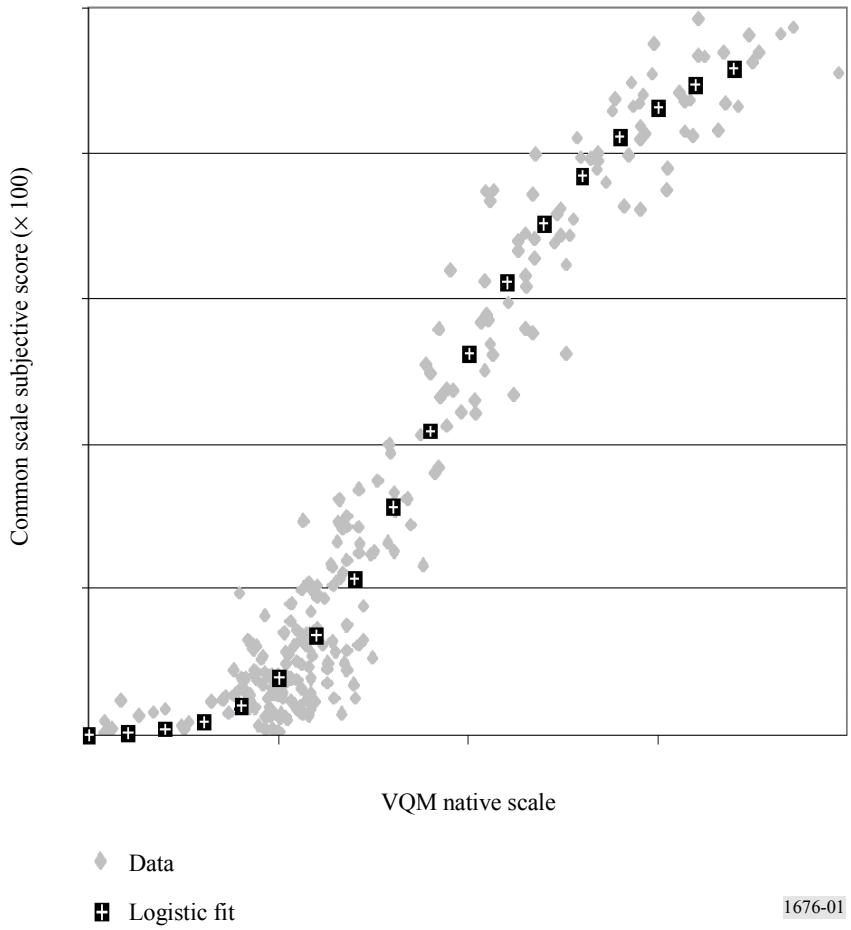
– monotonicity (the property of being either strictly increasing or strictly decreasing) over the specified domain of validity.

Of course, the fitting function would be most useful as a cross-calibration tool if it were monotonic over the entire theoretical domain of VQM scores, covered the entire subjective common scale from 0 to 1, and mapped to zero the VQM score that corresponds to a perfect video sequence

(no degradations, hence a null distortion). However, this ideal may not be attainable for certain VQMs and function families used to perform the fit.

FIGURE 1

Improved fit of data to VQM by mapping VQM to common scale



One possible family of fitting functions is the set of polynomials of order M. Another is a logistic function with the form:

$$\hat{O}_i = a + b / \{1 + c(O_i + d)^e\}$$

where a, b, c, d and e are fitting parameters. A third possibility is a logistic function with the form:

$$\hat{O}_i = a + (b - a) / \{1 + \exp[-c(O_i + d)]\}$$

where a, b, c, d are fitting parameters and $c > 0$. For convenience, we call these logistic forms Logistic I and Logistic II, respectively. The MATLAB code in Appendix 2 instantiates only a polynomial fit. Appendix 3 discusses possible methods of data fitting using the logistic functions.

The selection of a fitting-function family (including *a priori* setting of some of the parameters) depends on the asymptotic (best and worst) scores of the particular VQM.

The number of degrees of freedom used up by the fitting process is denoted by D. For example, if a linear fit is used, $D = 2$ since two free parameters are estimated in the fitting procedure. The fitting function that transforms objective VQM to the common scale is reported to facilitate industry comparison of two VQMs.

Once transformed to the common scale, any VQM can be cross-calibrated to any other VQM through the common scale. Representing the accuracy of a VQM in common scale facilitates comparisons between VQMs. Also, assuming the resolving power in the common scale does not vary much with the VQM score at which the resolving power is evaluated, the resolving power can be mapped through the inverse of the logistic function to the native scale. In the native scale, the ΔVQM from the common scale generates a VQM-score-dependent resolving power. A table or equation that provides such resolving powers (one at each VQM score in native scale) will have immediate meaning for users of the native scale.

2.3 METRIC 1: VQM accuracy based on statistical significance

We define a new quantitative measure of VQM accuracy, called resolving power, defined as the ΔVQM value above which the conditional subjective-score distributions have means that are statistically different from each other (typically at the 0.95 significance level). Such an "error bar" measure is needed in order for video service operators to judge the significance of VQM fluctuations.

Of several possible approaches to assessing a VQM's resolving power, the Student's *t*-test was chosen. This test was applied to the measurements in all pairs *i* and *j* of situations. Emerging from the test are the ΔVQM (i.e. the difference between the greater and lesser VQM score of *i* and *j*) and the significance from the *t*-test. This significance is the probability *p* that, given *i* and *j*, the greater VQM score is associated with the situation that has the greater true underlying mean subjective score. Thus, *p* is the probability that the observed difference in sample means of the subjective scores from *i* and *j* did not come from a single population mean, nor from population means that were ordered oppositely to the associated VQM scores. To capture this ordering requirement, the *t*-test must be one-tailed. For simplicity, the *t*-test was approximated by a *z*-test. This approximation is a close one when the number of viewers is large, as was the case for the Video Quality Experts Group (VQEG) data set (ITU-T Document COM 9-80-E).

An analysis of variance (ANOVA) test might seem better than the *t*-test method. However, although a single application of ANOVA will determine whether a statistical separation exists among a set of categories, further paired comparisons are needed to determine the magnitudes and conditions of the statistically significant differences. Also, ANOVA assumes equal category-data variances (which may not be true). Finally, although ANOVA resides in many software packages, finding the right software package may not be easy (e.g. not all ANOVA routines will accept different quantities of data in different categories).

The algorithm has the following steps:

Step 1: Start with an input data table with N rows, each row represents a different situation (i.e. a different source video and distortion). Each row i consists of the following: the source number, the distortion number, the VQM score O_i, the number of responses N_i, the mean subjective score $S_{i\bullet}$, and the sample variance of the subjective scores V_i.

Step 2: Transform the subjective scores $S_{i\bullet}$ to common scale $\hat{S}_{i\bullet}$ as described in § 2.1. The variance V_i of the subjective scores must also be scaled accordingly as:

$$\hat{V}_i = \frac{V_i}{(worst - best)^2}$$

Note that transforming the subjective scores and their variances are optional. It will not change the z statistic defined below, but it may change the VQM fitting process. Next, transform the VQM scores O_i to the common scale using a fitting function as discussed in § 2.2, and amplified in Appendix 3. The result of the fitting process is a set of common scale VQM scores $\hat{O}_i$. Display the coefficient values used in the fit, and also the VQM domain over which the fit was done (domain of validity).

Step 3: For each pair of distinct situations i and j ($i \neq j$), use a one-tailed z-test to assign a probability of significance to the difference between the greater and the lesser VQM ($\hat{O}_i$ and $\hat{O}_j$, respectively). The significance is the probability that the greater VQM score comes from the situation with the greater true underlying mean subjective score. The z score is:

$$z = \left(\hat{S}_{i\bullet} - \hat{S}_{j\bullet}\right) \Big/ \sqrt{\left(\hat{V}_i / N_i + \hat{V}_j / N_j\right)}$$

and the probability of significance of the z score $p(z)$ is just the cumulative distribution function of z:

$$p(z) = cdf(z) = (2\,\pi)^{-0.5} \int_{-\infty}^{z} \exp(-z^2/2)\ \mathrm{dz}$$

Step 4: Create a scatter plot of $p(z)$ (ordinate) versus ΔVQM score (abscissa). Given N situations, record each pair (i, j) with $i > j$, record the VQM difference $\hat{O}_i - \hat{O}_j$ in a vector of length $N(N-1)/2$ called ΔVQM (with index k), and record the corresponding z score in a vector called Z with length $N(N-1)/2$ (with the same index k). It is desired to ensure that ΔVQM(k) is always non-negative, which can be ensured by definition of the otherwise arbitrary ordering of the endpoints i and j. To ensure that this is so, if ΔVQM(k) is negative, then replace $Z(k)$ by $-Z(k)$ and ΔVQM(k) by $-$ΔVQM(k).

Step 5: Consider 19 bins (indexed by m) of ΔVQM, each one of which spans 1/10 the total range of ΔVQM. The bins overlap by 50%. Associate $\Delta\mathrm{VQM}_m$ with the midpoint of each bin and associate p_m with the mean of $p(z)$ for all z in bin m.

Step 6: Draw a curve through the points ($\Delta\mathrm{VQM}_m$, p_m), to produce a graph of p versus ΔVQM. Note that p can be interpreted as the average probability of significance.

Step 7: Select a threshold probability p, draw a horizontal line at the ordinate value p, and let its intercept with the curve of Step 6 determine the threshold ΔVQM, defined as the accuracy. For an average probability of significance of p or greater, the ΔVQM should exceed this threshold. Common choices of p are 0.68, 0.75, 0.90 and 0.95.

Having found a value of ΔVQM for a chosen p, one can use it directly in common scale – as would be appropriate for cross-calibration in Step 6. Alternatively, for other purposes, one has the option of inverse mapping this ΔVQM value back to the native scale to give a native scale resolving power R as a function of the native objective score O:

$$R(O) = \left| F^{-1}\left[F(O) + \Delta\text{VQM}\right] - O \right|$$

where F is the fitting function defined in § 2.2. For the logistic functions in § 2.2, the inverse of Logistic I is:

$$F^{-1}(x) = \left[(1/c)\,(b/[x-a]) - 1 \right]^{1/e} - d$$

and the inverse of Logistic II is:

$$F^{-1}(x) = d - 1\,(1/c)\,\text{In}\left[(b-a)/(x-a) - 1\right]$$

When $|\Delta\text{VQM}| << 1$, $R(O)$ can be approximated as:

$$RO = \left| \Delta\text{VQM} / F'(O) \right|$$

where $F'(O)$ is the derivative of F with respect to O. This approximation should suffice for most purposes.

NOTE 1 – For the logistic functions in § 2.2, the derivative of Logistic I is:

$$F'(x) = -bce\,(x+d)^{e-1} \Big/ \{1 + c(x+d)^{e}\}^2$$

and the derivative of Logistic II is:

$$F'(x) = c\,(b-a)\,\exp[-c(x-d)] \,/\, \{1 + \exp[-c(x-d)]\}^2$$

2.4 METRIC 2: VQM root-mean-squared error (RMSE) calculation

If the subjective data have roughly equal variance across the VQM scale, then a pooled estimate of variance, or resolving power, may be appropriate. As an example, we choose the RMSE. The basic idea behind the VQM RMSE calculation is to quantify the mean-squared error (MSE) between

fitted objective data and corresponding subjective data. The VQM RMSE between the fitted objective data $\hat{O}_i$ and the scaled subjective data $\hat{S}_{i\bullet}$ is computed as:

$$VQM\_RMSE = \sqrt{\frac{1}{N-D}\sum_{i=1}^{N}\left(\hat{O}_i - \hat{S}_{i\bullet}\right)^2}$$

where:

N: total number of situations (equal to IJ, where J is the number of scenes and I is the number of HRCs)

D: degrees of freedom used up by the objective to subjective curve fitting performed in § 2.2.

2.5 Classification plots

Classification errors are one way to evaluate the effectiveness of a VQM. A classification error is made when the subjective test and the VQM lead to different conclusions on a pair of data points. This section discusses the meaning of the classification errors, in terms of the plots of subjective z score versus ΔVQM described in the main text. For the following description, we use the common [0, 1] scale for both the subjective and objective scores. Here, "0" represents no impairment and "1" represents maximum impairment.

For any subjective test one can set a threshold Δz, that defines when two data points (A, B) are statistically equivalent and when they are statistically distinguishable[1]. Once this has been done, the subjective test results allow one to place each pair of data points (A, B) into one of three categories:

$\Delta z_{AB} < -\Delta z$ → A is better than B → *Bs*

$-\Delta z \le \Delta z_{AB} \le \Delta z$ → A is same as B → *Es*

$\Delta z < \Delta z_{AB}$ → A is worse than B → *Ws*

The abbreviations for the three categories (*Bs*, *Es* and *Ws*) denote subjectively better, subjectively equivalent, and subjectively worse, respectively.

Now consider a similar threshold for VQM values, Δo:

VQM (A) – VQM (B) $< -\Delta o$ → A is better than B → *Bo*

$-\Delta o \le$ VQM (A) – VQM (B) $\le \Delta o$ → A is same as B → *Eo*

$\Delta o <$ VQM (A) – VQM (B) → A is worse than B → *Wo*

The abbreviations for the three categories (*Bo*, *Eo* and *Wo*) denote objectively better, objectively equivalent, and objectively worse, respectively.

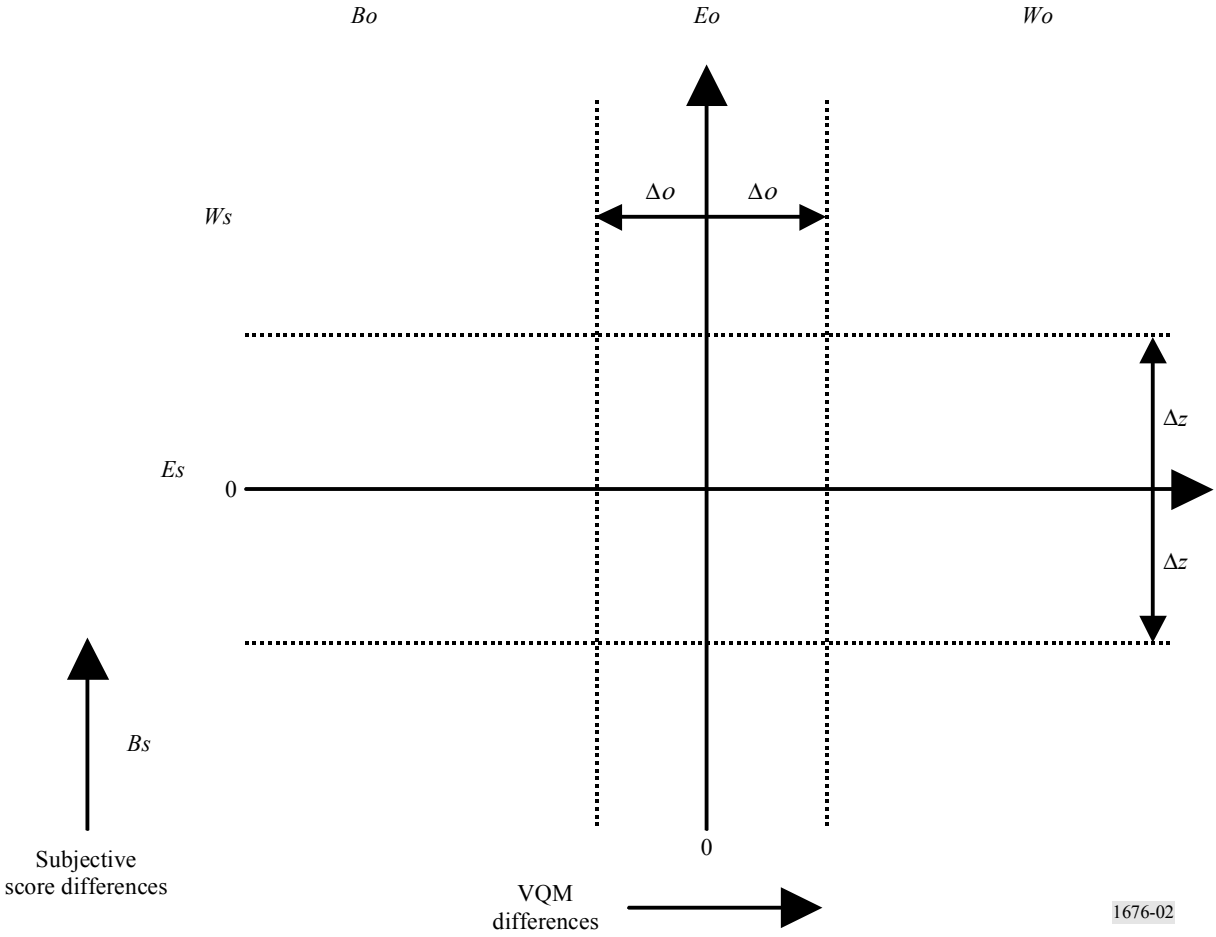
---

[1] The data points A and B actually represent sets of observations of two SRC/HRC combinations. As discussed in the main text, the quantity Δz_{AB} is the difference in the means of A and B ($\hat{S}_{A\bullet} - \hat{S}_{B\bullet}$), divided by the inferred standard deviation $\sqrt{(\hat{V}_A / N_A + \hat{V}_B / N_B)}$, where $\hat{V}_A$ is the variance of scores from situation A, and N_A is the number of observations from situation A, etc.

Since each pair of data points undergoes a three-way classification by the subjective test and a separate three-way classification by the VQM, there are nine possible outcomes. These nine outcome spaces are illustrated in Fig. 2 by the broken lines in the two-dimensional space of subjective-score difference versus VQM difference:

FIGURE 2

Schematic classification diagram



In the Table below, we label each of these nine outcomes with an eye towards answering the question "How does the VQM-based three-way classification compare with the subjective test-base three-way classification?".

	Bs	***Es***	***Ws***
Wo	False ranking	False differentiation	Correct decision
Eo	False tie	Correct decision	False tie
Bo	Correct decision	False differentiation	False ranking

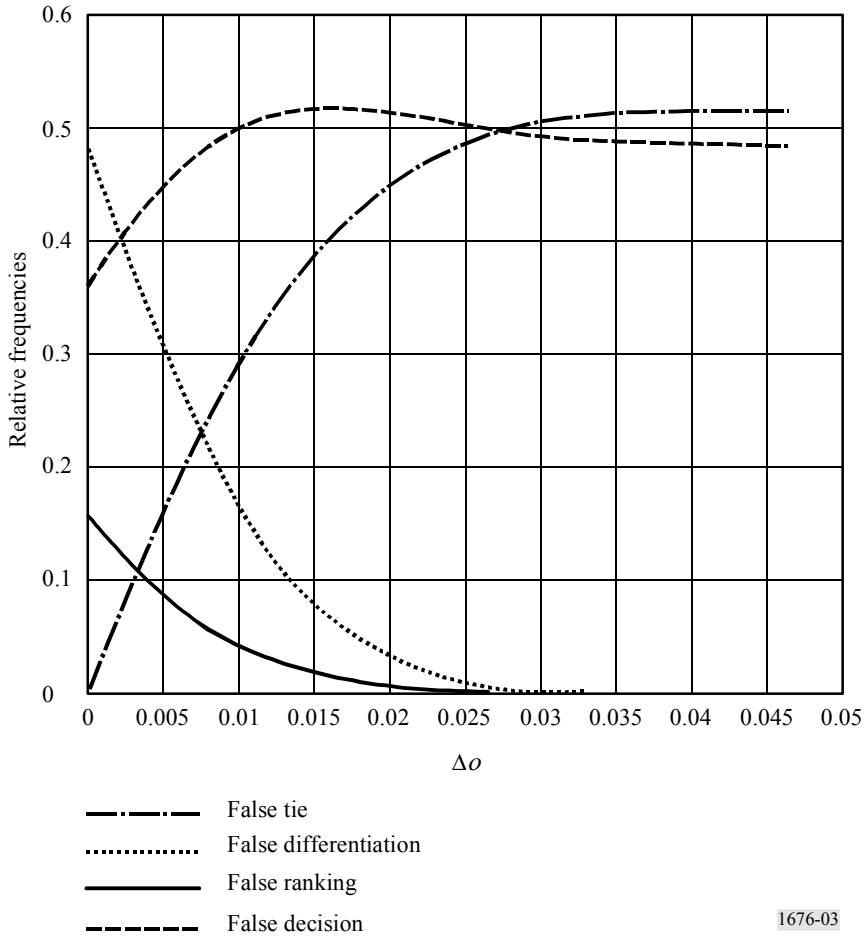
Note that for three of the outcomes, the VQM classification agrees with the subjective test classification. These three outcomes are labelled "correct decision". The six remaining outcomes correspond to three different types of errors that can arise when using a VQM. The false tie is

probably the least offensive error. This occurs when the subjective test says two data points are different but the VQM says they are the same. A false differentiation is usually more offensive. This occurs when the subjective test says two data points are the same but the VQM says they are different. The false ranking would generally be the most offensive error. In false ranking, the subjective test says *A* is better than *B*, but the VQM says *B* is better than *A*.

For any subjective test and any VQM, we can form all possible distinct pairs of data points and count the number of pairs that fall into each of the four distinct outcome categories: correct decision, false tie, false differentiation, and false ranking. We can then normalize by the total number of distinct pairs and report relative frequencies for these four outcome categories. In general these results will be functions of both Δs and Δo. Example results for a fictitious VQM are given in Fig. 3. Δz was selected to give an estimated 95% confidence in the subjective classifications and Δo is the free parameter on the *x*-axis of the graph.

FIGURE 3

Sample plot of frequencies of classification error



Note that as Δo is increased, the VQM will declare more and more pairs of data points as equivalent. This reduces the occurrences of false differentiations and false rankings, but increases the occurrence of false ties. As Δo goes to 0.05, the false-tie rate tends towards 0.52. At this point,

the VQM is declaring all pairs to be equivalent, and in doing so the VQM is wrong 52% of the time, and correct 48% of the time. This is consistent with the fact that, in this test, 48% of the pairs of data points were declared equivalent by the subjective test. One might use a graph like this to select an appropriate value of Δo. For example, one might select Δo to maximize the probability of making correct decisions, or one might select Δo to minimize some weighted sum of the error relative frequencies.

In the code that generated Fig. 3 (part of the MATLAB code in Appendix 2), the threshold used for the subjective test is subj_th. The threshold used for the ΔVQM, vqm_th, is left as a free parameter. The code plots the frequency of occurrence for the three different kinds of errors and for no error vs. vqm_th. An optimal value of vqm_th might be one that maximizes the frequency of occurrence of no error, or one that minimizes a cost-weighted sum of the errors. In general, it is likely that false ties will be the least offensive error, false differentiations will be more offensive, and false rankings will be the worst sort of error.

NOTE 1 – The nine outcomes and the three by three grid in (ΔVQM, subjective Z score) space is the most natural way to describe this analysis. This assumes bipolar values for ΔVQM. But the code has already taken the absolute value of ΔVQM (and replaced Z with $-Z$ for all points with negative values of ΔVQM). This does not change the mathematics, but the more natural description of the situation is now six outcomes and a 2 by 3 grid. Two correct outcomes (A better than B and A worse than B) have been folded on top of each other. There are still two false tie outcomes, but only one false differentiation outcome and one false ranking outcome.

3 Cross-calibrating two VQMs

The need to relate two VQMs is met by the transformation to a common scale described in § 2.1-2.2[2]. Once two VQMs (say, VQM1 and VQM2) are transformed to the common scale (through an agreed-upon subjective data set), the transformation from VQM1 to VQM2 is simply the forward transformation from VQM1 to the common scale and then the inverse transformation from common scale to VQM2. Models to be compared have to be referenced to a common data set. In cases for which the domains or ranges of the mapping mismatch, the cross-calibration must be declared to be undefined. This Recommendation does not specify a particular common data set.

---

[2] CAVEAT: One must use caution in making inferences from cross-calibration – e.g. cross-calibration of two VQMs does not mean one of the VQMs can be substituted error-free for the other. One reason for this limitation is that the present cross-calibration method depends on the particular subjective data that define the common scale. It might be argued that no subjective data is needed for a cross-calibration, and that one could connect two VQMs directly through their outputs given a particular set of inputs (trial and reference video pairs). However, no matter what set of VQM inputs are chosen for the cross-calibration, the VQMs may respond differently to some other videos. More fundamentally, even within the chosen input set, there are likely four inputs (1, 2, 3, 4) such that both VQM scores change in the same direction going from 1 to 2, but in opposite directions going from 3 to 4. Such behaviour is what makes one VQM better than another, and cannot be captured in any cross-calibration method.

Appendix 1
to Annex 1

Application of this Recommendation in the evaluation and validation of proposed VQMs

(Informative)

1 Elements of a full VQM disclosure

Each candidate VQM must be independently validated and fully disclosed such that it could be readily implemented by someone knowledgeable in the art. The description of newly proposed VQMs should include three different data sets:

– test vectors to check implementation of the VQM, including video inputs and resulting VQM outputs;

– validation/accuracy data, including subjective ratings and model outputs (spanning enough quality range to be representative of typical transmitted videos);

– data relating to other evaluation methods such as the Pearson linear correlation coefficient between objective and subjective scores, Spearman rank of order correlation between objective and subjective scores, and Outlier ratio.

Finally, there should be descriptions of scope and limitations, accuracy, and model cross-calibration as described in subsequent sections of this Recommendation.

2 Scope/limitations of a VQM

The scope of a VQM can include the following elements (an illustrative list, intended neither to be prescriptive nor exhaustive):

– the type of scene content (signal), e.g. high/low motion, colour versus black-and-white, interlaced versus progressive;

– the type and severity of artefacts (noise), driven by encoding techniques and bit rates (e.g. blurring, blockiness);

– the viewing conditions (including viewing distance, ambient illumination, and display parameters such as gamma, brightness and phosphor types).

Each VQM should be qualitatively assessed as to the type of scene content, type and severity of artefacts, and viewing conditions under which the VQM can and cannot operate effectively. It is important to list known problem areas (such as video distortions that include dropped frames) that would otherwise not be obvious, but the scope/limitations section is not intended to be an exhaustive list.

A set of four tables should be included in the description of the VQM's scope and limitations. The first three of these tables should enumerate all the distortions (HRCs) of the data set of the VQEG, and optionally others, as follows:

– a table of test factors, coding technologies, and applications for which the VQM has shown accuracy;

– a table of test factors, coding technologies, and applications for which the VQM has been tested but *not* shown the accuracy specified in § 2;

– a table of known test factors, coding technologies, and applications (including all of those used by VQEG) for which the VQM has not been tested, or where the VQM is not recommended.

In addition, there should be a table of test sequences used to determine test factors, coding technologies and applications for which the VQM has shown the accuracy specified in § 2.

Sample Tables are shown below.

TABLE 1

a) Test factors, coding technologies and applications for which the candidate VQM method has shown the specified accuracy

Bit rate	Resolution	Method	Comments
2 Mbit/s	3/4 resolution	mp@ml	This is horizontal resolution reduction only
2 Mbit/s	3/4 resolution	sp@ml	
4.5 Mbit/s		mp@ml	
3 Mbit/s		mp@ml	
1.5 Mbit/s	CIF	H.263	
768 kbit/s	CIF	H.263	
4.5 Mbit/s		mp@ml	Composite NTSC and/or PAL
6 Mbit/s		mp@ml	
8 Mbit/s		mp@ml	Composite NTSC and/or PAL
8 & 4.5 Mbit/s		mp@ml	Two codecs concatenated
19/PAL(NTSC)- 19/PAL(NTSC)- 12 Mbit/s		422p@ml	PAL or NTSC 3 generations
50-50-… –50 Mbit/s		422p@ml	7th generation with shift/I frame
19-19-12 Mbit/s		422p@ml	3rd generation
Not applicable		Not applicable	Multigeneration Betacam with drop-out (4 or 5, composite/component)

TABLE 1 (*end*)

b) Test factors, coding technologies and applications for which the VQM method has not shown the specified accuracy

Bit rate	Resolution	Method	Comments
4.5 Mbit/s		mp@ml	With errors
3 Mbit/s		mp@ml	With errors

These Tables exhaust the VQEG data set, so the sample Table 1c) would not contain any entries.

c) Test sequences used to determine test factors, coding technologies and applications for which the VQM has shown the specified accuracy

Sequence	Characteristics
Balloon-pops	Film, saturated colour, movement
NewYork 2	Masking effect, movement
Mobile&Calendar	Available in both formats, colour, movement
Betes_pas_betes	Colour, synthetic, movement, scene cut
Le_point	Colour, transparency, movement in all the directions
Autumn_leaves	Colour, landscape, zooming, water fall movement
Football	Colour, movement
Sailboat	Almost still
Susie	Skin colour
Tempête	Colour, movement

Appendix 2 to Annex 1

MATLAB Source Code

(Informative)

Below is a MATLAB subroutine called vqm_accuracy.m. This version scales the subjective data to [0, 1], applies a polynomial fit of the objective to the scaled subjective data, calculates all the metrics, and plots the VQM frequencies of “false tie”, “false differentiation”, “false ranking” and “correct decision”. It is sufficient to have Version 5.3.1 of MATLAB (1999) with the Statistics and Optimization toolboxes that are available separately. Software can also be developed that does not use either toolbox. The present code is intended as an illustrative example, and does not include all possible options and fitting functions.

Usage: At the matlab prompt, for VQM r0 type:

>load r0.dat

>vqm_accuracy(r0,-1,0,100,2)

For VQM r2, type:

>load r2.dat

>vqm_accuracy(r2,1,0,100,2)

Here, r0.dat and r2.dat are text files that contain a subset of the VQEG 525-line data. Each line in this file corresponds to a situation, and comprises an SRC number, an HRC number, VQM score, number of viewings, mean subjective score, and subjective-score variance. Once the r0 and r2 dat files are loaded, either form of vqm_accuracy may be run again.

In the first calling argument of vqm_accuracy, r0 corresponds to the PSNR model in TR A3, and r2 corresponds to the PQR model in TR A4. The second argument is 1 if the objective metric indicates worse image quality when it is larger, else the argument is –1. The third and fourth arguments are the nominal best and worst ratings on the native subjective scale. The final argument is the order of the polynomial to which the VQM is fit.

Source Code:

```
function vqm_accuracy (data_in, vqm_sign, best, worst, order)
% MATLAB function vqm_accuracy (data_in, vqm_sign, best, worst, order)
%
% Each row of the input data matrix data_in must be organized as
% [src_id hrc_id vqm num_view mos variance], where
%
%  src_id is the scene number
%  hrc_id is the hypothetical reference circuit number
%  vqm is the video quality metric score for this src_id x hrc_id
%  num_view is the number of viewers that rated this src_id x hrc_id
%  mos is the mean opinion score of this src_id x hrc_id
%  variance is the variance of this src_id x hrc_id
%
%  The total number of src x hrc combinations is size(data_in,1).
%
% vqm_sign = 1 or -1 and gives the direction of vqm with respect to
%      the common subjective scale. For instance, since "0" is
%      no impairment and "1" is maximum impairment on the common
%      scale, vqm_sign would be -1 for PSNR since higher values
%      of PSNR imply better quality (i.e., this is opposite to
%      the common subjective scale).
%
% mos and variance will be linarly scaled such that
%  best is scaled to zero (i.e., the best subjective rating)
%  worst is scaled to one (i.e., the worst subjective rating)
%
% order is the order of the polynomial fit used to map the objective data
% to the scaled subjective data (e.g., order = 1 is a linear fit).
%

% Number of src x hrc combinations
num_comb = size(data_in,1);
```

```
% Pick off the vectors we will use from data_in
vqm = data_in(:,3);
num_view = data_in(:,4);
mos = data_in(:,5);
variance = data_in(:,6);

% Scale the subjective data for [0,1]
mos = (mos-best)./(worst-best);
variance = variance./((worst-best)^2);

% Use long format for more decimal places in printouts
format('long');

% Fit the objective data to the scaled subjective data.
% Following code implements monotonic polynomial fitting using optimization
% toolbox routine lsqlin.
%
% Create x and dx arrays. For the dx slope array (holds the derivatives of
% mos with respect to vqm), the vqm_sign specifies the direction of the slope
% that must not change over the vqm range.
x = ones(num_comb,1);
dx = zeros(num_comb,1);
for col = 1:order
  x = [x vqm.^col];
  dx = [dx col*vqm.^(col-1)];
end
% The lsqlin routine uses <= inequalities. Thus, if vqm_sign is -1 (negative
% slope), we are correct but if vqm_sign is +1 (positive slope), we must
% multiple each side by -1.
if (vqm_sign == 1)
  dx = -1*dx;
end
fit = lsqlin(x,mos,dx,zeros(num_comb,1));
fit = flipud(fit)' % organize this fit same as what is output by polyfit

% vqm fitted to mos
vqm_hat = polyval(fit,vqm);

% Perform the vqm RMSE calculation using vqm_hat.
vqm_rmse = (sum((vqm_hat-mos).^2)/(num_comb-(order+1)))^0.5

% Perform the vqm resolution measurement on both vqm and vqm_hat.
vqm_pairs = repmat(vqm,1,num_comb)-repmat(vqm',num_comb,1);
vqm_hat_pairs = repmat(vqm_hat,1,num_comb)-repmat(vqm_hat',num_comb,1);
mos_pairs = repmat(mos,1,num_comb)-repmat(mos',num_comb,1);
stand_err_diff = sqrt(repmat(variance./num_view,1,num_comb)+ ...
  repmat((variance./num_view)',num_comb,1));
z_pairs = mos_pairs./stand_err_diff;

% Include everything above the diagonal.
delta_vqm = [];
delta_vqm_hat = [];
z = [];
for col = 2:num_comb
  delta_vqm = [delta_vqm; vqm_pairs(1:col-1,col)];
  delta_vqm_hat = [delta_vqm_hat; vqm_hat_pairs(1:col-1,col)];
  z = [z; z_pairs(1:col-1,col)];
end
```

```
% Switch on z and delta_vqm for negative delta_vqm
z_vqm = z;
negs_vqm = find(delta_vqm < 0);
delta_vqm(negs_vqm) = -delta_vqm(negs_vqm);
z_vqm(negs_vqm) = -z_vqm(negs_vqm);

z_vqm_hat = z;
negs_vqm_hat = find(delta_vqm_hat <0);
delta_vqm_hat(negs_vqm_hat) = -delta_vqm_hat(negs_vqm_hat);
z_vqm_hat(negs_vqm_hat) = -z_vqm_hat(negs_vqm_hat);

% Plot scatter plot of z_vqm versus delta_vqm in figure 1.
% Plot scatter plot of z_vqm_hat versus delta_vqm_hat in figure 2.
figure(1)
plot(delta_vqm,z_vqm,'.','markersize',1)
set(gca,'LineWidth',1)
set(gca,'FontName','Ariel')
set(gca,'fontsize',12)
xlabel('Delta VQM')
ylabel('Subjective Z Score')
grid on
print -dpng figure1

figure(2)
plot(delta_vqm_hat,z_vqm_hat,'.','markersize',1)
set(gca,'LineWidth',1)
set(gca,'FontName','Ariel')
set(gca,'fontsize',12)
xlabel('Delta VQM Hat')
ylabel('Subjective Z Score')
grid on
print -dpng figure2

% Plot average confidence that vqm(2) is worse than vqm(1) in figure 3.
% Plot average confidence that vqm_hat(2) is worse than vqm_hat(1) in
% figure 4. These are the resolving power plots.
%
% One control parameter for delta_vqm resolution plot; number of vqm bins
% equally spaced from min(delta_vqm) to max(delta_vqm).
% Sliding neighbood filter with 50% overlap means that there will actually
% be vqm_bins*2-1 points on the delta_vqm resolution plot.
cdf_z_vqm = .5+erf(z_vqm/sqrt(2))/2;
cdf_z_vqm_hat = .5+erf(z_vqm_hat/sqrt(2))/2;

vqm_bins = 10; % How many bins to divide full vqm range for local averaging
vqm_low = min(delta_vqm); % lower limit on delta_vqm
vqm_high = max(delta_vqm); % upper limit on delta_vqm
vqm_step = (vqm_high-vqm_low)/vqm_bins; % size of delta_vqm bins

vqm_hat_low = min(delta_vqm_hat);
vqm_hat_high = max(delta_vqm_hat);
vqm_hat_step = (vqm_hat_high-vqm_hat_low)/vqm_bins;

% lower, upper, and center bin locations
low_limits = [vqm_low:vqm_step/2:vqm_high-vqm_step];
high_limits = [vqm_low+vqm_step:vqm_step/2:vqm_high];
centers = [vqm_low+vqm_step/2:vqm_step/2:vqm_high-vqm_step/2];

hat_low_limits = [vqm_hat_low:vqm_hat_step/2:vqm_hat_high-vqm_hat_step];
hat_high_limits = [vqm_hat_low+vqm_hat_step:vqm_hat_step/2:vqm_hat_high];
hat_centers = [vqm_hat_low+vqm_hat_step/2:vqm_hat_step/2: ...
    vqm_hat_high-vqm_hat_step/2];
```

```
mean_cdf_z_vqm = zeros(1,2*vqm_bins-1);
mean_cdf_z_vqm_hat = zeros(1,2*vqm_bins-1);
for i=1:2*vqm_bins-1
  in_bin = find(low_limits(i) <= delta_vqm & delta_vqm < high_limits(i));
  hat_in_bin = find(hat_low_limits(i) <= delta_vqm_hat & ...
    delta_vqm_hat < hat_high_limits(i));
  mean_cdf_z_vqm(i) = mean(cdf_z_vqm(in_bin));
  mean_cdf_z_vqm_hat(i) = mean(cdf_z_vqm_hat(hat_in_bin));
end

% The x-axis is vqm(2)-vqm(1). For figure 3 (the vqm plot), if vqm_sign is
% 1, then the Y-axis is the average confidence that vqm(2) is worse than
% vqm(1). On the other hand, if vqm_sign is -1, then the Y-axis is the
% average confidence that vqm(1) is worse than vqm(2). Figure 4 is the plot
% for vqm_hat, and since it always has the same sign as mos, the Y-axis is
% always the average confidence that vqm_hat(2) is worse than vqm_hat(1).
if (vqm_sign == 1)
  figure(3)
  % VQM resolving power
  plot(centers,mean_cdf_z_vqm)
  grid
  set(gca,'LineWidth',1)
  set(gca,'FontName','Ariel')
  set(gca,'fontsize',11)
  xlabel('VQM(2)-VQM(1)')
  ylabel('Average Confidence VQM(2) is worse than VQM(1)')
  print -dpng figure3
else
  figure(3)
  % VQM resolving power
  plot(centers,1-mean_cdf_z_vqm)
  grid
  set(gca,'LineWidth',1)
  set(gca,'FontName','Ariel')
  set(gca,'fontsize',11)
  xlabel('VQM(2)-VQM(1)')
  ylabel('Average Confidence VQM(1) is worse than VQM(2)')
  print -dpng figure3
end

figure(4)
% VQM Hat resolving power.
plot(hat_centers,mean_cdf_z_vqm_hat)
grid
set(gca,'LineWidth',1)
set(gca,'FontName','Ariel')
set(gca,'fontsize',11)
xlabel('VQM Hat(2) - VQM Hat(1)')
ylabel('Average Confidence VQM Hat(2) is worse than VQM Hat(1)')
print -dpng figure4

% This portion of the code calculates and plots the relative frequencies of
% three types of classification errors. A classification error is made when
% the subjective test and the VQM lead to different conclusions on a pair
% of data points.
%
% Background: For any subjective test, one must set a threshold that will
% determine when two results are statistically equivalent, and when they are
% statistically distinguishable. Then for each pair of data points (A,B),
% the subjective test can yield one of three possible outcomes: (1) A better
% than B, (2) A same as B, and (3) A worse than B.
%
```

```
% If we define a similar threshold for VQM values, we have the same
% situation. For each pair of data points, VQM can yield one of three
% possible outcomes: (1) A better than B, (2) A same as B, and (3) A worse
% than B. Since each pair of data points undergoes three-way classification
% by the subjective test and three-way classification by the VQM, there are
% nine possible outcomes. For three of these outcomes, the subjective test
% and the VQM agree. If we take the subjective test to be correct by
% definition, and the VQM to be under test, then we say that for these three
% outcomes, the VQM is correct. In two other cases the VQM has committed the
% "false-tie" error (subjective test says A better than B, or A worse than B,
% but VQM says A same as B). In two other cases the VQM has committed the
% "false differentiation" error (subjective test says A same as B, but VQM
% says A better than B, or A worse than B.) Finally, there are two cases
% where the VQM has performed a false ranking (subjective test says A better
% than B, or A worse than B, but VQM says the opposite.) Thus, all nine
% outcomes are accounted for. Note that a three by three grid in
% (delta_vqm, subjective Z score) space describing the above could be drawn.
%
% In the code below, the threshold used for the subjective test is subj_th.
% The threshold used for the delta VQM is vqm_th and this is left as a free
% parameter. The code plots the frequency of occurrence for the three
% different kinds of errors and for no error vs. vqm_th. An optimal value of
% vqm_th might be one that maximizes the frequency of occurrence of no error,
% or one that minimizes a cost-weighted sum of the errors. Note that in
% general, it is likely that false ties will be the least offensive error,
% false differentiations will be more offensive, and false rankings will be
% the worst sort of error.
%
% For more details, see S. Voran, "Techniques for Comparing Objective and
% Subjective Speech Quality Tests," Proceedings of the Speech Quality
% Assessment Workshop, Bochum, Germany, November 1994.
%
% Note: The nine outcomes and the three by three grid in (delta_vqm,
% subjective Z score) space is the most natural way to describe this
% analysis. This assumes bipolar values for delta_vqm. But the code has
% already taken the absolute value of delta_vqm (and replaced Z with -Z for
% all points with negative values of delta_vqm). This does not change the
% math, but the more natural description of the situation is now 6 outcomes
% and a 2 by 3 grid. Two correct outcomes (A better than B and A worse
% than B) have been folded on top of each other. There are still two false
% tie outcomes, but only one false differentiation outcome and one false
% ranking outcome.

% Figure 5 is the plot for vqm and figure 6 is the plot for vqm_hat.
subj_th = 1.6; % 95 percent confidence
num_th = 50; % number of delta_vqm thresholds to examine
vqm_th_list = [vqm_low:(vqm_high-vqm_low)/num_th:vqm_high];
vqm_hat_th_list = [vqm_hat_low:(vqm_hat_high-vqm_hat_low)/num_th: ...
    vqm_hat_high];
rel_freqs = zeros(vqm_bins+1,4);
rel_hat_freqs = zeros(vqm_bins+1,4);
for i = 1:num_th+1
  vqm_th = vqm_th_list(i);
  vqm_hat_th = vqm_hat_th_list(i);
  % Number of data points in the false tie region
  rel_freqs(i,1) = length(find((delta_vqm < vqm_th) & ...
    (subj_th <= abs(z_vqm))));
  rel_hat_freqs(i,1) = length(find((delta_vqm_hat < vqm_hat_th) & ...
    (subj_th <= abs(z_vqm_hat))));
  % Number of data points in the false differentiation region
```

```
  rel_freqs(i,2) = length(find((vqm_th <= delta_vqm) & ...
    (abs(z_vqm) < subj_th)));
  rel_hat_freqs(i,2) = length(find((vqm_hat_th <= delta_vqm_hat) & ...
    (abs(z_vqm_hat) < subj_th)));
  % Number of data points in the false ranking region
  if (vqm_sign == 1)
    rel_freqs(i,3) = length(find((vqm_th <= delta_vqm) & ...
      (z_vqm <= -subj_th)));
  else
    rel_freqs(i,3) = length(find((vqm_th <= delta_vqm) & ...
      (z_vqm >= subj_th)));
  end
  rel_hat_freqs(i,3) = length(find((vqm_hat_th <= delta_vqm_hat) & ...
    (z_vqm_hat <= -subj_th)));
end
% Normalize counts by total number of points to get relative frequencies
rel_freqs = rel_freqs/length(z_vqm);
rel_hat_freqs = rel_hat_freqs/length(z_vqm_hat);
% Calculate relative frequency of correctness
rel_freqs(:,4) = (1-sum(rel_freqs(:,1:3)'))';
rel_hat_freqs(:,4) = (1-sum(rel_hat_freqs(:,1:3)'))';

% Figure 5 is plot for vqm and figure 6 is plot for vqm_hat.
figure(5)
% VQM Subjective Classification Errors
plot(vqm_th_list,rel_freqs(:,1),'m-.', vqm_th_list,rel_freqs(:,2),'r:', ...
  vqm_th_list,rel_freqs(:,3),'k-',vqm_th_list,rel_freqs(:,4),'b--');
grid
set(gca,'LineWidth',1)
set(gca,'FontName','Ariel')
set(gca,'fontsize',12)
xlabel('Delta VQM Significance Threshold')
ylabel('Relative Frequencies')
legend('False Tie','False Differentiation','False Ranking','Correct Decision')
print -dpng figure5

figure(6)
% VQM Hat Subjective Classification Errors
plot(vqm_hat_th_list,rel_hat_freqs(:,1),'m-.', ...
  vqm_hat_th_list,rel_hat_freqs(:,2),'r:', ...
  vqm_hat_th_list,rel_hat_freqs(:,3),'k-', ...
  vqm_hat_th_list,rel_hat_freqs(:,4),'b--');
grid
set(gca,'LineWidth',1)
set(gca,'FontName','Ariel')
set(gca,'fontsize',12)
xlabel('Delta VQM Hat Significance Threshold')
ylabel('Relative Frequencies')
legend('False Tie','False Differentiation','False Ranking','Correct Decision')
print -dpng figure6
```

Appendix 3
to Annex 1

Data-fitting to a common scale of VQM

(Informative)

As discussed in § 2.2 of the main text, the objective VQM data (O_i) are mapped to a new domain $\hat{O}_i = F(O_i)$. This domain is derived by fitting O_i to the scaled subjective data $(\hat{S}_{i\bullet})$ using a family of functions F (with fitting parameters) that have the properties of monotonicity and range mapping noted in § 5.2. The following are three alternative choices for the form of F, together with notes on data fitting using these functional forms.

1 Polynomial of order *M*

A polynomial that is fit to a set of data points is not guaranteed to be monotonic. The MATLAB optimization toolbox has a function lsqlin that ensures monotonicity over the extent of the data. However, monotonicity over the existing data domain does not ensure monotonicity over the entire theoretical domain (for example, 0 to infinity).

2 Logistic Function I

Fitting the objective VQM data (O_i) to the scaled subjective data $(\hat{S}_{i\bullet})$ can be done using a logistic function:

$$\hat{O}_i = F(O_i) = a + b/\{1 + c(O_i + d)^e\}$$

where a, b, c, d and e are fitting parameters. The fit function must be derived by non-linear least squares. (See MATLAB notes in the VQEG final Report[3].) The part of the function to be used is the monotonic part for $O > -d$ (hence constrain $d > -\min(O)$), and the s-curve shape appropriate to the data fit is ensured by constraining $e > 1$.

In certain cases, at least asymptotically, the perfect score in the native-scale objective model can be made to map to zero (the best score on the subjective scale), and the worst native-scale objective score possible can be made to map to the worst subjective score (unity, on the common scale). For example, consider the following case: Best objective score is zero, worst objective score is infinite. Here zero maps to zero and infinity maps to 1, so $a = 1$ and $b = -(1 + cd^e)$, hence:

$$F(O_i) = 1 - (1 + cd^e)/\{1 + c(O_i + d)^e\}$$

Fitting would take place on c, d, e, subject to d, $e > 0$.

[3] Constrained least squares non-linear curve fitting can be performed with the MATLAB function lsqcurvefit.

3 Logistic Function II

Fitting the objective VQM data (O_i) to the scaled subjective data ($\hat{S}_{i\bullet}$) can also be done using a logistic function:

$$\hat{O}_i = F(O_i) = a + (b - a)/\{1 + \exp[-c(O_i - d)]\}$$

where a, b, c and d, and $c > 0$ (ensured by defining $c = |\mathrm{C}|$ for real C). As with Logistic I, the fit function must be derived by non-linear least squares[4].

One might use this optimization in the case noted in § 2: Best objective score is zero, worst-objective score is infinite. Here zero maps to zero and infinity maps to 1, so $a = -\exp[-cd]$ and $b = -a \exp[cd]$. Hence:

$$F(O_i) = [1 - \exp(-c\, O_i)]/[1 + \exp\{c(d - O_i)\}]$$

Logistic Function II is also useful in the following case (which could arise when O_i is expressed in logarithmic coordinates such as decibels): Best objective score is infinite, worst objective score is negative-infinite. In that case infinity must map to 0, and negative infinity must map to 1. Hence $b = 0$, $a = 1$ and:

$$F(O_i) = 1/[1 + \exp\{c(O_i - d)\}]$$

---

[4] On page 31 of the VQEG final Report, the initial values for the parameters were chosen as a = maximum subjective score, b = minimum subjective score, c = 1, and d = mean objective score.